Vorlesung

Logik und Datenbanken

Nicole Schweikardt

Johann Wolfgang Goethe-Universität Frankfurt am Main

Sommersemester 2008

Konjunktive Anfragen (I)

2.1 Deskriptiver Ansatz: regelbasiert, graphisch und logikbasiert
2.2 Auswertungskomplexität
2.3 Algebraischer Ansatz: SPC-Algebra und SPJR-Algebra

Konjunktive Anfragen (I)

2.1 Deskriptiver Ansatz: regelbasiert, graphisch und logikbasiert
2.2 Auswertungskomplexität
2.3 Algebraischer Ansatz: SPC-Algebra und SPJR-Algebra

Regelbasierte Konjunktive Anfragen — Informell

Beispiel-Anfrage:

(4) Welche Kinos (Name & Adresse) spielen einen Film mit “George Clooney”?

Andere Formulierung:

Wenn es in *Filme* ein Tupel $\langle x_{Titel}, x_{Regie}, \text{“George Clooney”}\rangle$ und
in *Programm* ein Tupel $\langle x_{Kino}, x_{Titel}, x_{Zeit}\rangle$ und
in *Orte* ein Tupel $\langle x_{Kino}, x_{Adr}, x_{Tel}\rangle$ gibt,
dann nimm das Tupel $\langle x_{Kino},\ x_{Adr}\rangle$ in die Antwort auf

Als regelbasierte konjunktive Anfrage:

$$
\begin{aligned}
Ans(x_{Kino}, x_{Adr}) \ \leftarrow\ & Filme(x_{Titel}, x_{Regie}, \text{“George Clooney”}),\\
& Programm(x_{Kino}, x_{Titel}, x_{Zeit}),\\
& Orte(x_{Kino}, x_{Adr}, x_{Tel})
\end{aligned}
$$

Regelbasierte Konjunktive Anfragen — Präzise

Definition 2.1

- **var** sei eine abzählbar unendliche Menge von Variablen(symbolen), die disjunkt zu den Mengen **att**, **dom**, **relname** ist.
  (Einzelne Variablen bezeichnen wir i.d.R. mit $x, y, x_1, x_2, \ldots$)
- Ein Term ist ein Element aus **var** $\cup$ **dom**.
- Ein freies Tupel der Stelligkeit k ist ein Element aus $(\textbf{var} \cup \textbf{dom})^k$.

Definition 2.2

Sei **R** ein Datenbankschema.
Eine regelbasierte konjunktive Anfrage über **R** ist ein Ausdruck der Form

$$Ans(u) \leftarrow R_1(u_1), \ldots, R_\ell(u_\ell)$$

wobei $\ell \geqslant 0$, $R_1, \ldots, R_\ell \in \mathbf{R}$, $Ans \in \textbf{relname} \setminus \mathbf{R}$ und $u, u_1, \ldots, u_\ell$ freie Tupel der Stelligkeiten $arity(Ans), arity(R_1), \ldots, arity(R_\ell)$, so dass jede Variable, die in u vorkommt, auch in mindestens einem der Tupel $u_1, \ldots, u_\ell$ vorkommt.

Semantik regelbasierter konjunktiver Anfragen

Sei Q eine regelbasierte konjunktive Anfrage (über einem DB-Schema **R**) der Form

$$Ans(u) \leftarrow R_1(u_1), \ldots, R_\ell(u_\ell)$$

- Mit $Var(Q)$ bezeichnen wir die Menge aller Variablen, die in Q vorkommen.
- Eine Belegung für Q ist eine Abbildung $\beta : Var(Q) \to \mathbf{dom}$.

  Wir setzen β auf natürliche Weise fort zu einer Abbildung von $Var(Q) \cup \mathbf{dom}$ nach **dom**, so dass $\beta|_{\mathbf{dom}} = \text{id}$. Für ein freies Tupel $u = \langle e_1, \ldots, e_k \rangle$ setzen wir $\beta(u) := \langle \beta(e_1), \ldots, \beta(e_k) \rangle$.
- Der Anfrage Q ordnen wir die folgende Anfragefunktion $[\![Q]\!]$ zu:

$$[\![Q]\!](\mathbf{I}) := \left\{ \beta(u) \ : \ \begin{array}{l} \beta \text{ ist eine Belegung für } Q, \text{ so dass} \\ \beta(u_i) \in \mathbf{I}(R_i), \text{ f.a. } i \in \{1, \ldots, \ell\} \end{array} \right\}$$

  für alle Datenbanken $\mathbf{I} \in inst(\mathbf{R})$.

Beispiele (1/2)

- Die Anfrage (6) Welche (je 2) Regisseure haben schon in Filmen gespielt, in denen der jeweils andere Regie geführt hat?

  lässt sich durch folgende regelbasierte konjunktive Anfrage ausdrücken:

$$Antworten(x, y) \leftarrow Filme(z_1, x, y), Filme(z_2, y, x)$$

- Die Anfrage (5) Läuft zur Zeit ein “Detlev Buck” Film?

  lässt sich durch folgende regelbasierte konjunktive Anfrage ausdrücken:

$$Ans() \leftarrow Filme(x, \text{“Detlev Buck”}, y),\ Programm(z, x, w)$$

Ans ist hier also ein Relations-Name der Stelligkeit 0.
Erinnern Sie sich an unsere Konvention, dass die Ausgabe “$\emptyset$” der Antwort “nein” entspricht, und die Ausgabe der Menge $\{\langle\rangle\}$, die aus dem “Tupel der Stelligkeit 0” besteht, der Antwort “ja” entspricht.

Beispiele (2/2)

Betrachte die Datenbank $\mathbf{I} := \{\mathbf{I}(R), \mathbf{I}(S)\}$ mit
$\mathbf{I}(R) := \{\langle a,a\rangle, \langle a,b\rangle, \langle b,c\rangle, \langle c,b\rangle\}$ und $\mathbf{I}(S) := \{\langle d,a,b\rangle, \langle a,c,e\rangle, \langle b,a,c\rangle\}$.

- Die Anfrage $Q_1 :=$

  $$Ans_1(x_1,x_2,x_3) \leftarrow R(x_1,y), S(y,x_2,x_3)$$

  liefert auf $\mathbf{I}$ das Ergebnis $[\![Q_1]\!](\mathbf{I}) = \{\langle a,c,e\rangle, \langle a,a,c\rangle, \langle c,a,c\rangle\}$.

- Die Anfrage $Q_2 :=$

  $$Ans_2(x,y) \leftarrow R(x,z_1),\ S(z_1,a,z_2),\ R(y,z_2)$$

  liefert auf $\mathbf{I}$ das Ergebnis $[\![Q_2]\!](\mathbf{I}) = \{\langle a,b\rangle, \langle c,b\rangle\}$.

Bezeichnungen

Oft sagen wir kurz Regel, um eine regelbasierte konjunktive Anfrage zu bezeichnen.

Sei Q eine Regel der Form $Ans(u) \leftarrow R_1(u_1), \ldots, R_\ell(u_\ell)$

- $Ans(u)$ wird als Kopf der Regel bezeichnet.
- $R_1(u_1), \ldots, R_\ell(u_\ell)$ wird als Rumpf der Regel bezeichnet.
- Die Relations-Namen aus **R** werden als extensionale Datenbankprädikate (kurz: edb-Prädikate) bezeichnet.
  Wir schreiben $edb(Q)$, um die Menge aller edb-Prädikate zu bezeichnen, die in Q vorkommen.
- Der Relations-Namen, der im Kopf von Q vorkommt, wird als intensionales Datenbankprädikat (kurz: idb-Prädikat) bezeichnet.
- Mit $adom(Q)$ bezeichnen wir die Menge aller Konstanten (also Elemente aus **dom**), die in Q vorkommen. ("*adom*" steht für "active domain")

Der “Active Domain” einer Datenbank

Definition 2.3

Sei **R** ein Datenbankschema und sei **I** eine Datenbank vom Schema **R**.

Der Active Domain von **I**, kurz: $adom(\mathbf{I})$, ist die Menge aller Elemente aus **dom**, die in **I** vorkommen. D.h.:

$$adom(\mathbf{I}) \;=\; \bigcup_{R\in\mathbf{R}} adom\big(\mathbf{I}(R)\big)$$

wobei für jedes R aus **R** gilt: $adom\big(\mathbf{I}(R)\big)$ ist die kleinste Teilmenge von **dom**, so dass jedes Tupel $t \in \mathbf{I}(R)$ eine Funktion von $sort(R)$ nach $adom\big(\mathbf{I}(R)\big)$ ist.

Ist Q eine Anfrage und **I** eine Datenbank, so setzen wir

$$adom(Q,\mathbf{I}) \;:=\; adom(Q) \cup adom(\mathbf{I})$$

Proposition 2.4

*Für jede regelbasierte konjunktive Anfrage Q und jede Datenbank **I** (vom passenden DB-Schema) gilt:* $adom\big(\llbracket Q\rrbracket(\mathbf{I})\big) \;\subseteq\; adom(Q,\mathbf{I})$.

Beweis: siehe Tafel.

Monotonie und Erfüllbarkeit

Sind **I** und **J** zwei Datenbanken vom gleichen Schema **R**, so sagen wir "**J** ist eine Erweiterung von **I**" und schreiben kurz "**I** $\subseteq$ **J**", falls für alle $R \in$ **R** gilt: **I**$(R) \subseteq$ **J**(R) (d.h. jedes Tupel, das in einer Relation von **I** vorkommt, kommt auch in der entsprechenden Relation von **J** vor).

Definition 2.5

Sei **R** ein DB-Schema und sei q eine Anfragefunktion über **R**.

(a) q heißt monoton, falls für alle Datenbanken **I** und **J** über **R** gilt:
Falls **I** $\subseteq$ **J**, so ist $q(\mathbf{I}) \subseteq q(\mathbf{J})$.

(b) q heißt erfüllbar, falls es eine Datenbank **I** gibt mit $q(\mathbf{I}) \neq \emptyset$.

Satz 2.6

Jede regelbasierte konjunktive Anfrage ist monoton und erfüllbar.

Beweis: siehe Tafel.

Anwendung von Satz 2.6

Satz 2.6 liefert ein einfaches Kriterium, um zu zeigen, dass bestimmte Anfragefunktionen nicht durch eine regelbasierte konjunktive Anfrage beschrieben werden können:

> Wenn eine Anfragefunktion q nicht monoton ist, dann kann sie auch nicht durch eine regelbasierte konjunktive Aussage beschrieben werden.

Beispiel: Die Anfrage

(15) Welche Filme laufen nur zu 1 Uhrzeit?

ist nicht monoton, kann also nicht durch eine regelbasierte konjunktive Anfrage beschrieben werden.

Vorsicht: Dies heißt nicht, dass jede monotone Anfragefunktion durch eine Regel beschrieben werden kann!

"Graphische" Variante: Tableau-Anfragen

Beispiel-Anfrage:

(4) Welche Kinos (Name & Adresse) spielen einen Film mit "George Clooney"?

Darstellung als Tableau **T** (ähnlich zu QBE):

Filme

Titel	Regie	Schauspieler
x_{Titel}	x_{Regie}	"George Clooney"

Programm

Kino	Titel	Zeit
x_{Kino}	x_{Titel}	x_{Zeit}

Orte

Kino	Adresse	Telefon
x_{Kino}	x_{Adr}	x_{Tel}

Zugehörige Tableau-Anfrage: $\big(\mathbf{T},\ \langle x_{Kino},\ x_{Adr}\rangle\big)$

Tableaus — Präzise

Definition 2.7

Sei $\mathbf{R}$ ein Datenbankschema und R ein Relationsschema.

- Ein Tableau über R (auch: Einzel-Tableau) ist eine endliche Menge von freien Tupeln (also Tupeln über $\mathbf{dom} \cup \mathbf{var}$) der Stelligkeit $arity(R)$.

  (D.h. ein Tableau über R ist eine "Relation vom Schema R, die als Einträge nicht nur Elemente aus **dom**, sondern auch Variablen aus **var** haben kann".)

- Ein Tableau $\mathbf{T}$ über $\mathbf{R}$ ist eine Abbildung, die jedem $R \in \mathbf{R}$ ein Tableau über R zuordnet.

  (D.h. ein Tableau über $\mathbf{R}$ ist eine "Datenbank vom Schema $\mathbf{R}$, die als Einträge auch Variablen enthalten kann".)

- Eine Tableau-Anfrage über $\mathbf{R}$ (bzw. R) ist von der Form $(\mathbf{T}, u)$, wobei $\mathbf{T}$ ein Tableau über $\mathbf{R}$ (bzw. R) und u ein freies Tupel ist, so dass jede Variable, die in u vorkommt, auch in $adom(\mathbf{T})$ vorkommt.

  u heißt Zusammenfassung der Anfrage $(\mathbf{T}, u)$.

Semantik von Tableau-Anfragen

Sei $Q = (\mathbf{T}, u)$ eine Tableau-Anfrage.

- $Var(Q)$ bezeichnet die Menge aller Variablen, die in u oder $\mathbf{T}$ vorkommen. $adom(Q)$ bezeichnet die Menge aller Konstanten, die in u oder $\mathbf{T}$ vorkommen.
- Eine Belegung für Q ist eine Abbildung $\beta : Var(Q) \to \mathbf{dom}$.
- Sei $\mathbf{I}$ eine Datenbank vom Schema $\mathbf{R}$.

  Eine Belegung β für Q heißt Einbettung von $\mathbf{T}$ in $\mathbf{I}$, falls "$\beta(\mathbf{T}) \subseteq \mathbf{I}$", d.h. f.a. $R \in \mathbf{R}$ gilt:

$$\beta\big(\mathbf{T}(R)\big) \ := \ \{\beta(t) : t \in \mathbf{T}(R)\} \ \subseteq \ \mathbf{I}(R).$$

- Der Tableau-Anfrage Q ordnen wir die folgende Anfragefunktion $[\![Q]\!]$ zu:

$$[\![Q]\!](\mathbf{I}) \ := \ \{\beta(u) : \beta \text{ ist eine Einbettung von } \mathbf{T} \text{ in } \mathbf{I}\}$$

  für alle Datenbanken $\mathbf{I} \in inst(\mathbf{R})$.